

Corporate Information Network (COIN): The Fraunhofer IESE Experience Factory

Andreas Jedlitschka, Klaus-Dieter Althoff, Björn Decker, Susanne Hartkopf, Markus Nick

Fraunhofer Institute Experimental Software Engineering (IESE),
Sauerwiesen 6, D-67551 Kaiserslautern
Germany
{jedl|althoff|decker|hartkopf|nick}@iese.fhg.de

Abstract

Experience-based continuous learning is essential for improving products, processes, and technologies in emerging as well as established areas of business and engineering science. It can be facilitated by case-based organizational learning, meaning that relevant experience is captured in the form of cases for reuse in a corporate experience repository. For obvious reasons, learning from experience needs to be a permanent endeavor. Thus, an organization has to handle a “continuous stream of experience.” For this purpose, an “Experience Factory” was established at Fraunhofer IESE, with the COIN initiative. The objectives of COIN are to provide users with valuable information/knowledge at the right time, in an adequate representation, and within the actual context (“just-in-time”).

1. Introduction

In all emerging areas of business and engineering science, there is normally a lack of explicit knowledge about their underlying processes, products, and technologies. Usually, such knowledge is built up through individual learning from the experience of the people involved. The field of organizational learning tries to increase the effectiveness of individual human learning for a whole organization. Besides improving internal communication (group learning), organizational learning also includes documenting relevant knowledge and storing it (for reuse) in an organizational/corporate memory (Abecker et al 1998, van Heijst, van der Speck, and Kruizinga 1996).

An approach known from software engineering called Experience Factory (EF) (Basili, Caldiera, and Rombach 1994) goes one step further. Knowledge (in the form of processes, products, and technologies) is enriched by explicitly documented experience (e.g., lessons that were learned during the practical application of the knowledge). The EF approach includes collecting, documenting, and

storing of such experience as experience packages in an experience base (EB), which is an organizational memory for relevant knowledge and experience. This tries to make human “learning from experience” explicit in order to further support repository-based organizational learning. We call this experience-factory-based organizational learning.

With in this paper we first give an example for an operative EF (Sec. 2), then looking toward enhancements of the existing approach. These enhancements deal with new strategies for information providing (Sec. 3) including (a) aggregation and adaptation and (b) information “pushing”, followed by concepts for flexible “in time” information gathering (Sec. 4), and maintenance of EB content (Sec. 5). Finally a short outlook to future work is given (Sec. 6).

2.COIN: An Example of an Experience Factory

Celebrating its fifth anniversary in early 2001, our institute grew to 120 employees in the last years. One characteristic of such fast-growing organizations is the small number of employees who have been in the organization from the beginning. Therefore, this small group of experts becomes a scarce resource as information providers. Hence, it is important (a) to provide the less experienced people with default processes and guidelines to jump-start them and (b) to facilitate experience sharing among them to build up their expertise more quickly. Since the size of our institute does not allow to talk to all people on a weekly basis, experience sharing on a personal basis does not work. Therefore, a project named COIN (Corporate Information Network) was launched. Additionally, COIN is used as a real project environment for the development and validation of technologies and methods for goal-oriented experience management including knowledge elicitation, processing, dissemination, presentation, maintenance, and evaluation. It consists of three main parts: the EB, the COIN team, and an intranet representation.

Within the EB included in COIN, all kinds of experience necessary for our daily business are stored (e.g., projects, business processes, document templates, guidelines, observations, improvement suggestions, problems that occurred and problem fixes that were applied). Defined

processes (structured interviews within project touch-down meetings) populate this EB systematically with experience typically needed by our project teams. Dedicated improvement processes analyze problems that have occurred, devise improvement actions to avoid their recurrence, and implement strategic decisions by the institute's leadership. However, elicitation, distribution, and integration of process descriptions and lessons learned need an investment of effort (Decker et al. 2001). The project teams using the process descriptions and gaining the experiences cannot be expected to invest this effort. Compared to the objectives of the organization, projects have a short-term perspective, focusing on the development goals of the project. Therefore, an organizational unit, which is responsible for knowledge management is required. This organizational unit has to be separated from the project teams. According to (Basili, Caldiera, and Rombach 1994, Althoff, Birk, and Tautz 2000), this separate organizational unit is called EF, which for the IESE is operationalized by the COIN team (Tautz 2000).

The current focus is on two major subject areas: business process descriptions and lessons learned. The lessons learned are in the form of guidelines, observations, and problems. The guidelines act as solutions or mitigation strategies for the problems. An observation describes the results of an application of a guideline. Besides this, many different kinds of experience like artifacts developed during projects are to be stored in the EB. Each is called an experience package. In addition, these experience packages are highly interrelated. For example, projects produce deliverables in the form of slide presentations and reports. Slide presentations may be summaries of reports. Observations and problems are gained during a project while a particular business process was performed, that is, we have to deal with context-sensitive experience. Such kind of experience is unique in the sense that the same context will not recur. Therefore, people will be searching for experience that has been gained in *similar* contexts. Both, the requirement for supporting different kinds of interrelated experience packages and the need for context-sensitive, similarity-based retrieval, demand a specialized technical infrastructure for the EB.

These are common requirements for an EB (Tautz 2000). Our solution to meet these requirements is INTERESTS (Intelligent Retrieval and Storage System) (Althoff et al. 1999). It consists of a general purpose browser for accessing and presenting the EB contents using a standard web browser, an EB server synchronizing (and logging) access to the EB, and a commercial case-based reasoning (CBR) (Althoff 2001) tool (CBR-Works/oreng from tec:inno, Germany; e.g. (Schulz 1999), which is used for the actual EB. Each experience package is implemented as a "case" based on a structural CBR approach (Bergmann et al. 1999). This includes a domain ontology for modeling the different types of case concepts, formal and informal case attributes together with the respective similarity measures, as well as relations between cases.

Within an experiment the benefits of this EB approach have already been demonstrated (Tautz 2000). Until now we have gathered nearly two years of operational experience in maintaining COIN, and we have successfully adapted COIN to partners/customers. Based on this experience we have widened the requirements of COIN towards an organization-wide information and knowledge management system. Other applications not yet considered (e.g. human resource and educational systems) may deliver valuable information, too. Additional information can lead to a more precise and better aggregation and adaptation of knowledge to users needs, but also requires the integration of the respective applications.

The support of our employees within projects shall be improved (a) with more flexible and faster mechanisms for sharing information and (b) by moving from a "pull" to a "push" strategy in the sense of providing the right information at the right time. Additionally, an open architecture with an enhanced technical infrastructure enabling the integration of and communication between applications will be developed (Decker and Jedlitschka 2001).

3. New Strategies for Providing Information/Knowledge

3.1. Aggregation and Adaptation of Information

Every member of an organization or, more abstract, every role has different needs with regard to the granularity of information. Stepping higher on the organizational or project level, information has to be aggregated and adapted more and more. The user gets the aggregated and adapted information in addition with an attribute, telling him about the degree of utility (personalized or evaluated experience) and the name of the author. On demand detailed information is available.

While project members need specific and in-depth information about their status within the project, the project leader is more interested in an overview of all project activities. For him it is valuable information that a deviation will occur because of illness of a project member. Experiences available dealing with those cases, regarding the risk plan, can assist him in evaluating the critical potential of this state. If he detects a business-critical state, the information is forwarded on a "red-phone" channel to the respective persons.

Further research work has to be done here.

3.2. "Push" of Information/Knowledge

We are developing an improved user interface, which provides a single point of access. With his login in combination with stored user data (his organizational role, project roles, and skills) and a chosen view (concrete project) the user provides the actual context, for example:

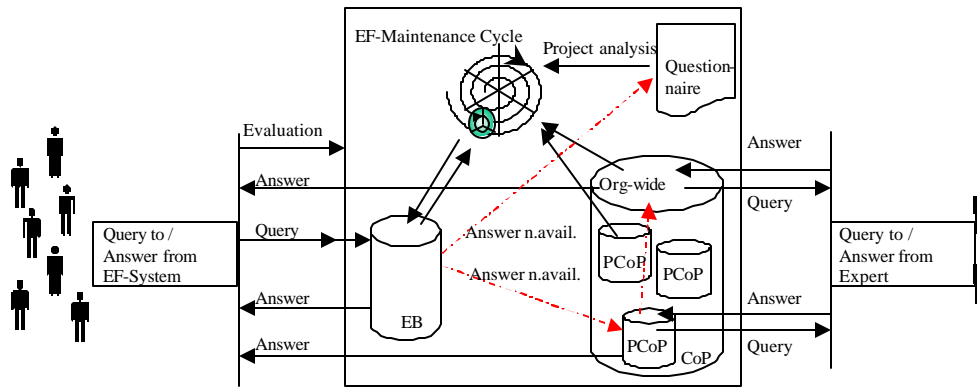


Fig. 1: Correlation: Experience base (EB) and communities of practice (CoP)

“role: developer; project: x; task: code testing” (the task is determined from the project plan). The given context is used to deliver knowledge collected within similar contexts without an explicit user query (“push” of information). The user can ignore this but, hopefully, he will at least evaluate the usefulness of the delivered information within his actual context.

Additionally, he can send specific queries to the EB (“pull”) (see Fig.1).

4. Community of Practice Base (CoP)

To get information, current users have to send a query to the EB. As practice shows, sometimes there is no appropriate case or experience package available for the specified problem. The user has to find his own solution, which tends to be available only to a very small group of people, unless he tells the COIN team about the gained experience. Currently project experiences are collected periodically and at the end of a project using project analysis interviews (i.e., a structured interview for acquiring lessons learned from project members). The project members tell their experiences within those interviews to the EF team, whose members are responsible to extract and derive lessons learned in form of guidelines, observations, and problems and to put them into the EB. For some problems, occurring within the projects this process is too slow. To present a solution, we are aiming at extending the EF through a more flexible concept, namely communities of practice (CoP).

CoP handle specific problems for which there is no information in the EB available, so far. In such a case the query is, with the agreement of the user, forwarded to the project-specific community of practice (PCoP) (see Fig.1). Every project member who currently has got a view on this project, will see the question nearly at the same time. They can assist by providing their own experience and simultaneously, they extend the knowledge base. Intuitively, the CoP supports the collection of tacit and personal knowledge. If after a while (the asking user can give a deadline) nobody answered the question (sufficiently), it is sent to the organization-wide CoP (if the

user agrees), where every user can answer the question. In addition, it could become one of the duties of some very experienced IESE members to look up the CoP at least once a week. The asking user should be able to evaluate the utility of the given answer according to his specific context by giving bonus points. These points can be gathered and an award like the “Expert of the Month/Year” can be instantiated, which is expected to motivate people to use this feature of COIN.

To support project analyses in a more specific way, such questions can also extend the questionnaire for the interview. The project member who sends the query now should be able to answer the question, because of experiences that solved the task, told by others through the CoP or made by himself. In this context it seems to be important to mention that the collection of both positive *and* negative experiences is necessary in case of a knowledge network (Bartsch-Spörl, Jargon, and Althoff 2001). An approach to archive project-specific CoPs within the project will be developed to avoid loss of experience.

5. Maintenance

Another part of the operative work of an EF besides the collection of experiences during project analysis is the maintenance of the EB content. With every new input to the EB, existing experience packages can be confirmed or questioned (Nick, Althoff, and Tautz 2001). This work will be supported by the introduction of the utility evaluation by the users. The EF Maintenance cycle (see Fig.1) shall symbolize the necessary activities. Rejected content can be discussed and widely evaluated using the CoP. At any rate, questions not yet answered or rejected content have to be considered as hints for maintenance, that is, in-depth analyses. The results will help to improve (a) EB content but also (b) information aggregation and adaptation, which includes education of the agents through the user by “carrot (bonus points) and the stick (rejection)”. Future COIN will support EF maintenance with automatically generated hints about users’ evaluation, general usage and, especially, the kind of system usage.

6. Outlook

In the future, we do not want to burden users with overhead for searching information or asking for experience. We are looking towards a solution for a single point of access to all knowledge and information produced in an organization, only restricted by access rights defined by (a) the organization in form of the employee's role within it, (b) the projects and the according role the employee plays and (c) the owner of a piece of information. The knowledge and information offered shall be aggregated and adapted based on experiences to users' actual needs, determined by the actual context. New tools like the CoP will help to explicate more valuable knowledge and information residing tacitly in experts' heads. Our vision is to provide users with valuable information/knowledge at the right time, in an adequate representation and within the actual context ("just-in-time"). Further work still to be done includes information adaptation, complete instantiation of the current COIN architecture as well as an approach for its (continuous) improvement.

References

- Abecker, A.; Bernardi, A.; Hinkelmann, K.; Kühn, O.; Sintek, M.: Towards a Technology for Organizational Memories, IEEE Intelligent Systems 1998.
- Althoff, K.-D.; Birk, A.; Hartkopf, S.; Müller, W.; Nick, M.; Surmann, D. Tautz, C.: Managing Software Engineering Experience for Comprehensive Reuse; Proceedings of the Eleventh Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, June 1999; Knowledge Systems Institute; Skokie, Illinois, USA; 1999.
- Althoff, K.-D.; Bomarius, F.; Tautz, C.: Knowledge Management for Building Learning Software Organizations, Information Systems Frontiers 2:3/4, 349-367, 2000; Kluwer Academic Publishers
- Althoff, K.-D. (2001). Case-Based Reasoning. To appear in: S.K. Chang (Ed.), Handbook on Software Engineering and Knowledge Engineering. World Scientific (40 pages)
- Basili, V.R.; Caldiera, G.; Rombach, D.: Experience Factory; In Marciniak, J.J. ed., Encyclopedia of Software Engineering, vol 1, 469-476; John Wiley & Sons; 1994.
- Bergmann, R., Breen, S., Göker, M., Manago, M., and Wess, S. (1999). *Developing Industrial Case-Based Reasoning Applications - The INRECA-Methodology*. Springer Verlag, LNAI 1612
- Bartsch-Spörl, B., Jargon, C. & Althoff, K.-D. (2001). Wissensmanagement in der Praxis - Erfahrungen aus erfolgreichen und weniger erfolgreichen WM-Projekten. Workshop auf der 1. Konferenz Professionelles Wissensmanagement: Erfahrungen und Visionen, Baden-Baden 14.-16. März 2001, (<http://demolab.iese.fhg.de:8080/AK-Praktisches-Wissensmanagement/WM2001-WS5/>)
- Decker, B.; Althoff, K.-D.; Nick, M.; Tautz, C.: Integrating Business Process Descriptions and Lessons Learned with an Experience Factory; In Professionelles Wissensmanagement - Erfahrungen und Visionen; Aachen 2001; Shaker Verlag S.54-58
- Decker, B.; Jedlitschka, A.: The Integrated Corporate Information Network iCoiN: A Comprehensive, Web-based Experience Factory; IESE-Report 031.01/E; Fraunhofer IESE; Kaiserslautern; 2001
- Nick, M., Althoff, K.-D. & Tautz, C. (2001). Systematic Maintenance for Corporate Experience Repositories. Computational Intelligence 17(2), 364-386
- Tautz, C.: Customizing Software Engineering Experience Management Systems to Organizational Needs; Ph. D. diss., Dept. of Computer Science, University of Kaiserslautern, Germany; 2000; Stuttgart: Fraunhofer IRB Verlag
- Schulz, S. 1999. CBR-Works - a state-of-the-art shell for case-based application building. In *7th German Workshop on Case-Based Reasoning*, Wuerzburg, Germany.
- van Heijst, G., van der Speck, R., and Kruizinga, E. 1996. Organizing corporate memories. In *Proceedings of the Tenth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, SRDG Publications, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4.